A Game Theoretic Model of a Protocol for Data Possession Verification[*]

Nouha Oualha, Pietro Michiardi and Yves Roudier
Institut Eurécom, Sophia Antipolis, France
{oualha, michiardi, roudier}@eurecom.fr

Abstract

This paper discusses how to model a protocol for the verification of data possession intended to secure a peer-to-peer storage application. The verification protocol is a primitive for storage assessment, and indirectly motivates nodes to behave cooperatively within the application. The capability of the protocol to enforce cooperation between a data holder and a data owner is proved theoretically by modeling the verification protocol as a Bayesian game, and demonstrating that the solution of the game is an equilibrium where both parties are cooperative.

1. Introduction

The capabilities of today's multimedia enabled mobile devices together with the need to ubiquitously (and durably, contrary to peer-to-peer file-sharing) access one's own data result in an increasing interest in peer-to-peer data storage. Data storage applications are evolving from a trusted infrastructure (e.g., as in OceanStore [1]) to a self-organized architecture, the larger scale of the system making it necessary to implement data management services such as distributed data storage in a cooperative fashion. Since cooperation between nodes is not guaranteed, data are exposed to new threats. Beyond malicious attacks, in which nodes purely aim at disrupting the storage service, self-organization results in a new form of denial of service called selfishness: nodes may discard some data they promised to store for other nodes in order to gain resource for their own usage. Because of the scattered nature and dynamics of storage, assessing data availability is not an immediate operation, contrary to observing packet forwarding for instance. Several works have designed cryptographic verification protocols in order to assess data possession (e.g., [2], [3], [4], and [5]). However, not all of these approaches address selfishness, and those who do so do not model cooperation.

We proposed a verification protocol for such peer-to-peer storage applications based on probabilistic challenge-responses between the data owner node and the data holder node. This protocol makes it possible for the data owner to react to the destruction of stored data, as well as it implicitly motivates data holders to keep the data in their storage space. The contribution of this paper is the validation of this security primitive with respect its cooperation enforcement function for data storage. We use game theory to model remuneration incentives based on such verifications.

The remainder of the paper is organized as follows. Section 2 describes the protocol for verifying data possession. Section 3 introduces the one-stage Bayesian games that model it. Section 4 then presents the repeated game that provides a more realistic model of our protocol. These two sections illustrate how the perfect Bayesian equilibrium, a solution of the game, validates the verification protocol.

2. Probabilistically verifying data possession

We consider a data storage application in which a node may cooperatively store its personal data at another node, taking advantage of the excess storage space offered by the latter node. Most approaches to distributed storage set on periodically verifying if data holder nodes still possess the data they have stored, thereby relying on a verification of data possession.

Such a verification protocol generally consists of challenges with which the data owner node (called O thereafter) regularly probes the data holder node (called H). However, this periodic validation comes at an additional communication and computational cost. We suggested in [6] to address this concern through the use of a probabilistic verification instead of a deterministic one as do most of existent proposals.

The verification protocol summarized in Figure 1 comprises two phases:

[*] This work was supported by the GET Initiative program on autonomic and spontaneous networks, the PACALAB project, and the ReSIST IST NoE.

Storage phase: The data is split into n indexed chunks, which are then encrypted. Additionally, the integrity of chunks is protected by "signing" the data: digital signature may be used although expensive, but may be replaced by less expensive methods like DES symmetric encryption [7] with a secret key, or keyed one-way hash function such as HMAC [8], or even concentric encryption. Protected chunks are sent to H.

Challenge phase: O randomly chooses one index corresponding to a data chunk (the probability to choose this index the next time does not change) and sends it to H. H answers with the corresponding chunk and its signature. O verifies the validity of the signature (and then, the chunk is of course deleted). Since chunk indexes are chosen randomly, H should keep all chunks stored to answer correctly all possible challenges from O. The challenge operation is repeated periodically until either O retrieves its data or it detects that H has destroyed a data chunk.

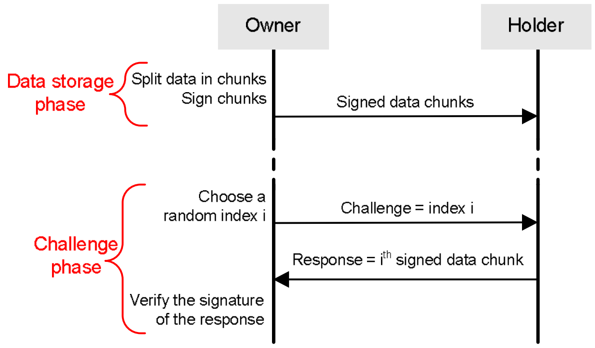

Figure 1 Data storage and challenge phases

The verification protocol only requires O to have the public key for the signature, or the secret key if some form of hashing is used. The challenge phase only verifies one data chunk, which makes it easier to handle for environments with scarce resources.

In this protocol, the selfish node may be depicted as a node destroying a portion of data chunks, as example the holder node destroys k chunks over the n chunks, $r=k/n$. So, the holder can answer correctly to challenges with probability $(1-r)$.

The following sections analyze how the proposed verification protocol helps enforcing cooperation.

3. Game theoretic model

The verification protocol described above is modeled using the analytical framework of game theory [9]. Game theory provides a language to describe, analyze, and understand strategic scenarios. In our model we assume the presence of two players (also termed actors) involved in the strategic process of deciding whether to cooperate or not on one hand, and to punish or reward on the other hand.

Our game models how incentives can be built based on the regular verification of the correct storage of data, as promised by holders. Cooperation incentives are expressed as remunerations: H is rewarded for a correct response while it is charged when responding incorrectly. The game however does not model nodes that decline storage requests from data owners, nor any case of a cooperative peer transferring data it stores to other peers when it plans a forthcoming disconnection.

The outcome of this modeling is the validation of the existence of cooperation equilibria after a series of probabilistic verifications, and the evaluation of the parameters to be taken into account to design proper incentives. Two games are introduced that respectively model the holder's strategy and the owner's strategy.

3.1. Game elements

The essential elements of our model are:

- Players: the individuals who make decisions: data owner and data holder. A player is assumed "rational", i.e., a player is a participant in the game and whose goal is to choose the actions that produce his most preferred outcomes.
- Payoffs: the numeric values assigned to the outcomes produced by the various combinations of actions. Payoffs represent the preference ordering of players over the outcomes.
- Information: information set for a player summarizes what the player knows when it gets to make a decision.
- Chance: probability distribution over chance events. We represent chance events by a random move of *nature* which is a pseudo-player whose actions are purely mechanical and probabilistic.

3.2. Game models

The storage protocol is modeled as a Bayesian game. In such a game, information about the characteristics of other players is incomplete, and nature is introduced as a player for modeling uncertainty.

Figure 2 illustrates the structure of our one-stage game in the extensive form (in the form of a tree where there is a complete description of how the game is played over time). A one-stage game corresponds to the phase of one challenge conducted by O towards H. Notations used in figures are explained in Table 1.

The parameters G, R, R' and D, in Table 1, are measured in the same units, e.g., the number of data bytes or data chunks stored. Also regarding data stored in a distributed fashion, we presume that the remote storage space has more value than local storage space, which explains that *G*>*R*>*D*.

Table 1 Notations

Notations		Explication
Players	O	data owner
	H	data holder
Errors	M	malfunction of H
	N	normal function of H
Types	C	H is cooperative
	S	H is selfish
	F	H is faulty
Signals	s	succeed O's challenge
	f	fail O's challenge
Actions	rw	reward H
	fg	do not do anything
	pn	punish H
Payoffs	*G*	distributed storage gained by O
	D	supplementary storage provided by H
	R	reward charge, such that *R*>*S*>0
	R'	punishment charge, such that *G-R*>*R'*>0
Chance	*q*	probability of challenge's success for a selfish holder H
	d	Probability of hardware failure (for H)

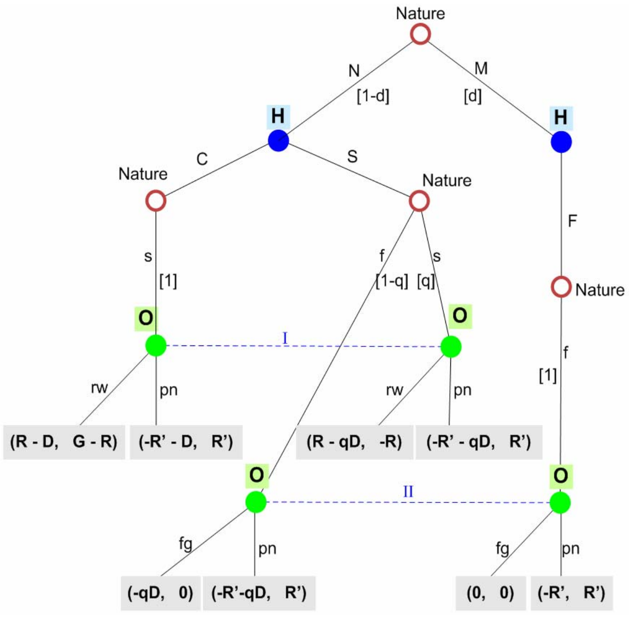

Figure 2 Modeling the holder strategy

The game (depicted in Figure 2) models the fact that the holder H may follow two possible strategies, or in game theoretical terms, be of two types: cooperative, that is, it will store an owner's data until its retrieval; or selfish, that is, it will destroy data chunks with probability 1-*q*. These types are respectively referred as "C" and "S". If H chooses the type "C", it succeeds in answering a challenge requested by O as modeled by the emission of signal "s". However, it may fail because of a hardware crash or error for instance, which occurs with probability *d*, and is modeled by the emission of signal "f". The failure to answering a challenge is either an incorrect response to the challenge or, more frequently, no response at all (after some time-out). If H chooses type "S", it may successfully answer a challenge only with a probability equal to *q*(1-*d*). Otherwise, it will behave like a faulty peer. In addition, real faults may still happen with probability *d*. the owner O is not informed about H's type, which is why O cannot distinguish between "C" and "S" despite the fact that H's signal is seen by O. Such situations that cannot be discriminated belong to the same so-called "set of information". The two sets of information I and II depicted in the game diagram correspond respectively to success and failure signals.

In this paper, we will consider a simplified version of the game of Figure 2, in which the risk of hardware failure for H is simply neglected (*d*=0). This simplification allows easier computations in the next sections, while focusing on holder strategies.

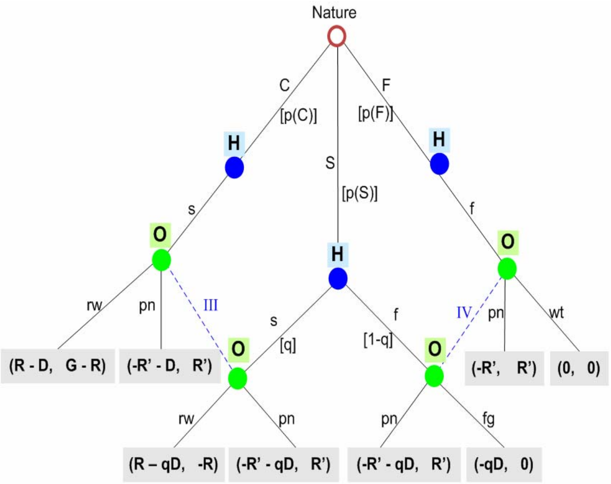

Figure 3 Modeling the owner strategy

The game model of Figure 2 is a sequential game with asymmetric distribution of information, since the holder H is informed about its type, but the owner O is not informed. However, O can probabilistically determine H's type based on its prior beliefs, such beliefs typically reflecting H's reputation. With every verification performed, O updates its beliefs according to Bayes' formula. To describe O's prior beliefs about H's type, we derive a second game model depicted in Figure 3. This model is a typical signaling game, that is, players have asymmetric information. The game is modeling the owner strategy: the game will use signals based on H's type as determined by the Nature. H, the informed player, has different types given by nature;

while H knows its type, O does not. Based on the knowledge of its own type, H sends signals which O can observe but which do not provide perfect information about H's type. In our model for instance, the set of information III may describe a cooperative or selfish H, and the set IV may describe a selfish or faulty H.

3.3. Equilibria

The solution of the game, which constitutes player's best response to the actions of the other player, is called an equilibrium. The following sections define the Nash equilibrium and the perfect Bayesian equilibrium of the game.

Nash Equilibrium: Nash Equilibrium is the set of players' strategy choices where no player can benefit by changing its strategy while the other player keeps its strategy unchanged. To define the Nash equilibrium of the game, the normal form of the game of Figure 2 (which lists each player's strategies and the payoffs that result from each possible combination of choices) is presented below in Table 2.

Table 2 Normal form of the game of Figure 2

		O	
		rw	pn
H	C	(*R-D*, *G-R*)	(*-R'-D*, *R'*)
	S	(*R-qD*, - *R*)	(*-R'-qD*, *R'*)

We assume that *G-R* > *R'*. If H chooses the type "C", then O, by strict dominance, chooses the action "rw" because the payoff associated to "rw" (=*G-R*) is higher than the payoff associated to "pn" (=*R'*). By choosing "rw", the better response by H is "S" because *R-D*<*R-qD*, and so, O will prefer to choose "pn" because *R'*>0>-*R*. At this point, neither O or H can have a benefit by changing to another strategy. So, ("S", "pn") is a Nash equilibrium. The normal form game leads to an equilibrium where non-cooperation is the best response for players.

Compared to the extensive form game, the normal form game lacks the information on whether O is informed or not about the type of H. The view of incomplete information is not represented within the normal form. Another equilibrium, the perfect Bayesian equilibrium, takes into account this view.

Perfect Bayesian Equilibrium. A perfect Bayesian equilibrium is a strategy profile σ*=(σ_1*,σ_2*) and posterior beliefs $\mu(\cdot|m)$ such that:

1. $\forall$type t, $\sigma_1^* \in \arg\max_{\sigma_1}(U_1(\sigma_1,\sigma_2^*,t))$
2. $\forall$signal m, $\sigma_2^* \in \arg\max_{\sigma_2}(\sum_t \mu(t\,|\,m)U_2(m,\sigma_2,t))$
3. $\mu(t\,|\,m) = p(t)\times\sigma_1^*(m\,|\,t)\big/\sum_{t'} p(t')\times\sigma_1^*(m\,|\,t')$

Finding the perfect Bayesian Equilibrium of the game means finding the following probabilities ([10]):

$\sigma_1^*(s\,|\,C)=1$ $\quad\sigma_1^*(f\,|\,C)=0$

$\sigma_1^*(s\,|\,S)=q$ $\quad\sigma_1^*(f\,|\,S)=1-q$

$\sigma_1^*(s\,|\,F)=0$ $\quad\sigma_1^*(f\,|\,F)=1$

$\sigma_2^*(rw\,|\,s)=u_1$ $\quad\sigma_2^*(fg\,|\,s)=v_1=0$ $\quad\sigma_2^*(pn\,|\,s)=1-u_1=w_1$

$\sigma_2^*(rw\,|\,f)=u_2=0$ $\quad\sigma_2^*(fg\,|\,f)=v_2$ $\quad\sigma_2^*(pn|\,f)=1-v_2=w_2$

Thus, the belief update equations are as follows:

$\mu(C\,|\,s)=p(C)/(p(C)+p(S)\times q)$

$\mu(S\,|\,s)=p(S)\times q/(p(C)+p(S)\times q)$

$\mu(F\,|\,s)=0$ $\quad\mu(C\,|\,f)=0$

$\mu(S\,|\,f)=p(S)\times(1-q)/(p(S)\times(1-q)+p(F))$

$\mu(F\,|\,f)=p(F)/(p(S)\times(1-q)+p(F))$

H's payoffs corresponding to each type is given by:

$\mathrm{U}_1(\sigma_1,\sigma_2^*,\ C)=u_1\times(R+R')-R'-D$

$\mathrm{U}_1(\sigma_1,\sigma_2^*,\ S)=q\times[u_1\times(R+R')+R'\times w_2-R'-D]-R'\times w_2$

$\mathrm{U}_1(\sigma_1,\sigma_2^*,\ F)=-R'\times w_2$

Expected O's payoffs for each signal sent by H is given by:

$$\sum_t \mu(\mathrm{t}\,|\,\mathrm{s})\mathrm{U}_2(s,\sigma_2,\ t)=u_1[Gp(C)/(p(C)+p(S)q)-R-R']+R'$$

$$\sum_t \mu(\mathrm{t}\,|\,\mathrm{f})\mathrm{U}_2(f,\sigma_2,\ t)=R'w_2$$

There are two case solutions:

Case 1: if $G\times p(C)/(p(C)+p(S))-R-R'\geq 0$, then σ_2* is maximized for u_1=1 and w_2=1. Because *R*+*R'*-*D*>0, σ_1* is maximized for *q*=1. The perfect Bayesian equilibrium is the strategy where:

$\sigma_1^*(s\,|\,S)=1$ $\quad\sigma_1^*(f\,|\,S)=0$

$\sigma_2^*(rw\,|\,s)=1$ $\quad\sigma_2^*(fg\,|\,s)=0$ $\quad\sigma_2^*(pn\,|\,s)=0$

$\sigma_2^*(rw\,|\,f)=0$ $\quad\sigma_2^*(fg\,|\,f)=0$ $\quad\sigma_2^*(pn\,|\,f)=1$

$p(S)/p(C)\leq(G-R-R')/(R+R')$

The equilibrium of the game leads to a strategy where O and H cooperate.

Case 2: if $G\times p(C)/(p(C)+p(S))-R-R'<0$, then σ_2* is maximized for w_2=1 only. The choice of u_1 is dependent on *q* and vice versa. If u_1=0, then σ_1 is maximal for *q*=0, and for *q*=0, σ_2 is maximal for u_1=1, and for u_1=1, σ_1 is maximal for q=1, however, for q=1, σ_2 is maximal for u_1=0, and so on. There is no perfect Bayesian equilibrium for this case.

4. Repeated game

We analyze a class of repeated games in which the informed player's type is persistent and the history of actions is perfectly observable. This context rightly represents the periodic iteration of the verification protocol performed by the owner node to assess whether the holder node is still storing the data it promises to store. The analyzed repeated game is the game of Figure 2 and Figure 3 iterated while maintaining H's type. These games are played for finite times, but no player knows the exact game termination time. The probability p captures the probability of "natural" termination of the repeated game (e.g., loss of connection between O and H). Additionally, the owner node O has the possibility to stop the repeated game if it detects the selfishness or the failure of H (H is of type "S" or "F"). The payoff at the i^{th} period is designated by $g_i=(g_i^H, g_i^O)$. The sum of per-period payoffs is given by:

$$g = (\sum_{i=0}^{\infty} (1-p)^i g_i^H, \sum_{i=0}^{\infty} (1-p)^i g_i^O)$$

4.1. Action profiles

From the signals sent per-period by H, O may infer the type of H. There are three distinct possible action profiles:

1. (s, rw), (s, rw), (s, rw), …
2. (f, pn)
3. (s, rw), (s, rw), …, (s, rw), (f, pn)

At the first round, if the signal is "f", O infers that the type of H is either "S" or "F", for both cases it is better to play the action "pn". If the signal is "s", then, the best response of O is to play "rw". If the signal changes to "f", O concludes that H is of type "S" and the action played is "pn".

4.2. Numerical evaluation of the repeated game framework

The games of Figure 2 and Figure 3 are iterated and evaluated within different scenarios. The evaluation is performed using a custom simulator, and games' parameters are measured in MB (Mega Bytes) unit (1 MB=106 bytes). The evaluated scenarios permit to define additional requirements on the values of the reward returned R, the punishment charge R', and the impact of the probability p on the cooperativeness of the holder.

At first, we consider the repeated game of Figure 2. H chooses the strategy that maximizes its payoff. To make H choose the type "C" over "S", its outcome by choosing "C" must be higher than its outcome choosing "S". For this, the reward R must be bigger than a minimum value. R must verify a minimum threshold to motivate the cooperation of H, this minimum is depicted in Figure 4. This figure shows as well that it is possible to confine the incentives to rewarding the cooperative player. But then, the minimum threshold for the reward R increases.

The beneficial impact for H when increasing the probability p (q low) is illustrated in Figure 5. We notice that for p roughly bigger than 0.6, H must strategically choose to be selfish with a ratio q very low (=0.1), however, for p less than 0.6, a high ratio q (=0.9) is better for H. This demonstrates that the repetition of the game motivates the holder H to cooperate (q approximating 1).

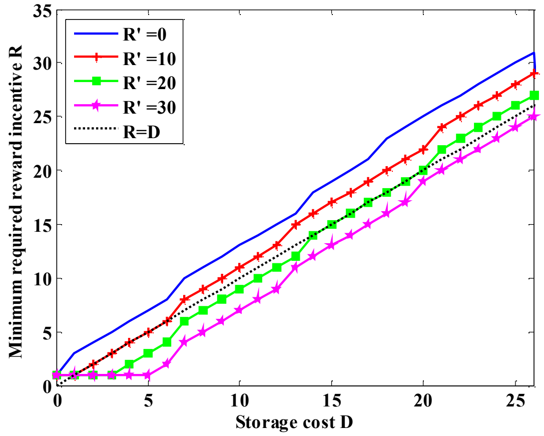

Figure 4 Minimum threshold for reward *R* over storage cost *D*, for various values of punishment incentive *R'*. *G*=30, *p*=0.2, *q*=0.5.

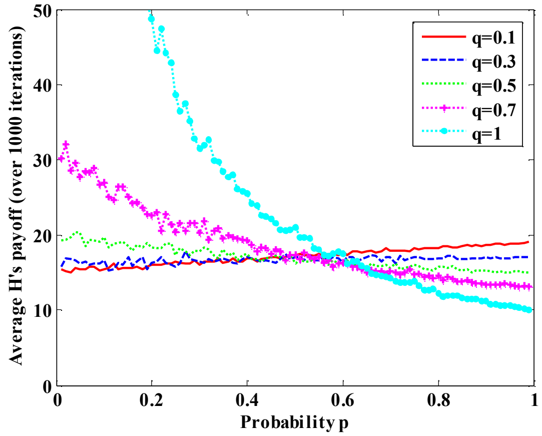

Figure 5 Average selfish H payoffs over the probability *p*, for various values of *q*. *G*=30, *R*=20, *R'*=5, *S*=10.

At this point, we consider the repeated game of Figure 3. In the previously studied one stage game, we put forward the inequality, for which we have a perfect Bayesian equilibrium where both O and H cooperates, $p(S)/p(C) \le (G-R-R')/(R+R')$. However for the repeated game, this inequality does not exactly hold. Figure 6 demonstrates that there exists a given ratio of prior beliefs p(S)/p(C) such that above this ratio O must not cooperate (i.e., O must stop the game by playing the action "pn", for punishment). For example, for the given particular game' parameters, if p(S)/p(C)=4 (e.g., p(S)=0.4 and p(C)=0.1), O must play "pn" because its average payoff if it plays other than "pn" is less than *R'*=5.

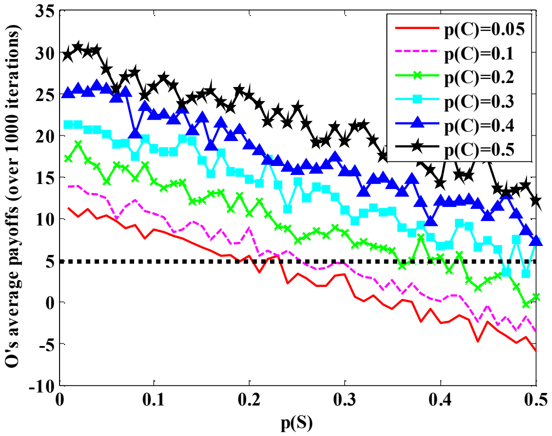

Figure 6 Average O's payoffs over prior beliefs on type "S", varying prior beliefs over type "C". *G*=30, *R*=20, *R'*=5, *D*=10, *p*=0.2, *q*=0.5.

4.3. Summary

The repeated game of Figure 2 represents an interaction between a data owner and a data holder from a data holder perspective. For this repeated game, we aim to encourage the cooperation of the holder by making its cooperative behavior the best strategically choice to make. For this, we showed the inequalities that the reward *R* and the punishment *R'* should verify. We demonstrated as well that it is possible to restrict the incentives to simply rewarding the holder (*R'*=0). Besides, the result on the probability *p* shows that iteration of the game favors the cooperativeness of H. On the other hand, the repeated game of Figure 3 illustrates the interaction of a data owner with a holder from the owner perspective. For this repeated game, we aim, this time, to guide the owner in choosing the best response to holder actions based on the prior beliefs about this very holder. These prior beliefs correspond to holder reputation. Using numerical results, it is possible to define which actions the owner must follow for a given ratio p(S)/p(C).

5. Conclusion

In this paper we proposed a verification protocol as a means to construct a periodic cooperation detection mechanism in the context of peer-to-peer distributed storage for wireless ad hoc networks. The protocol was designed to indirectly motivate nodes to behave cooperatively. The inherent incentive compatibility property of the proposed protocol was theoretically validated using a game theoretical model. We showed that the Bayesian game model of our protocol allows solutions where both parties of the game are cooperative.

As future work, we plan to construct the whole mechanism of cooperation enforcement, and to validate the mechanism by means of realistic simulation scenarios and results obtained from a real experimentation.

6. References

[1] J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao. OceanStore: An architecture for globalscale persistent storage. In Proceedings of the Ninth international Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2000), November. 2000.

[2] M. Lillibridge, S. Elnikety, A. Birrell, M. Burrows, and M. Isard, "A Cooperative Internet Backup Scheme", *In Proceedings of the 2003 Usenix Annual Technical Conference (General Track), pp. 29-41*, San Antonio, Texas, June 2003.

[3] G. Caronni and M. Waldvogel. Establishing Trust in Distributed Storage Providers. In *Third IEEE P2P Conference*, Linkoping, March, 2003.

[4] Y. Deswarte, J.-J. Quisquater, and A. Saïdane. Remote Integrity Checking. *In Proceedings of Sixth Working Conference on Integrity and Internal Control in Information Systems (IICIS)*, 2004.

[5] D. G. Filho, P. S. L. M. Barreto. Demonstrating data possession and uncheatable data transfer. In *IACR Cryptology ePrint Archive*, 2006.

[6] N. Oualha and Y. Roudier. Probabilistically secure cooperative distributed storage. Technical Report RR-07-188, Institut Eurécom, February 2007.

[7] National Bureau of Standards. Data Encryption Standard. *Federal Information Processing Standards Publication No. 46*, , January 15th, 1977.

[8] M. Bellare, R. Canetti, and H. Krawczyk. HMAC: Keyed-Hashing for Message Authentication. *RFC 2104*, Internet Engineering Task Force, February 1997.

[9] Theodore L. Turocy and Bernhard von Stengel. Game theory. *Cdamresearch report lse-cdam-2001-09*, London School of Economics, October 2001.

[10] Farhad Ghassemi. Signaling games, 2006. Available: http://www.cs.ubc.ca/~kevinlb/teaching/cs532a%20-%202006/Projects/FarhadGhassemi.pdf.